Exhibit C

EPMR

A program for crystallographic molecular replacement by evolutionary search.

Version 2.5

Authors: Charles R. Kissinger, Daniel K. Gehlhaar, & Bradley A. Smith
Agouron Pharmaceuticals, Inc.

Contents

Introduction

EPMR is a program that finds crystallographic molecular replacement solutions using an evolutionary search algorithm. The program directly optimizes the three rotational and three positional parameters for the search model with respect to the correlation coefficient between Fo and Fc. The program operates as follows:

- An initial set of random solutions (random orientations and positions for the search model) is generated.
- The correlation coefficient is calculated for each orientation.
- A fraction of the highest scoring orientations are retained and used to regenerate a complete set of new trial orientations. This is done by applying random alterations to the orientation angles and translations for each 'surviving' solution.
- The correlation coefficients for the new population are calculated, the population is again regenerated from the top scoring solutions, and this procedure is repeated for a number of cycles.
- At the end of this evolutionary optimization, a traditional conjugate gradient minimization is performed. This is simply a local, rigid-body refinement of the search molecule.

The use of an evolutionary algorithm allows for efficient searching of the six-dimensional space. It is several orders of magnitude faster than a brute-force, systematic search.

The program calculates structure factors very quickly using the method of Huber and Schneider [J. Appl. Cryst. (1985) 18, 165-169]. A traditional structure factor calculation is done only once - for the search model set at the origin of a P1 cell. Subsequent structure factor calculations are done by transforming reflection indices according to the rotations and translations applied to the model and the relationship between the P1 and real cells, interpolating into the grid of P1 structure factors and summing over the symmetry operators of the crystal. This is much faster than an FFT calculation.

The program finds a single 'solution' for the search model in each run. In cases where you are looking for multiple molecules in the asymmetric unit, you must look for each solution sequentially. This can be done in two different ways. You can have the program find the first solution and write it out. You then enter it as partial structure in subsequent runs. Or you can direct the program to find the first solution, keep it as partial structure, and keep going to look for as many solutions as you like. In this way, multi-body searches can be totally automatic.

Because of the stochastic nature of the optimization process, you will not get the correct solution on every run, even with a very good search model. The success rate is USUALLY very high assuming an adequate model, but you probably want to try at least 10 runs if you have a difficult problem. For search models that are poor and at the limit of detectability, the search efficiency can be quite low. If you have a molecular replacement problem that has not yielded a solution by any other means, a reasonable last resort is to set up EPMR to do as many runs as your patience and computing resources will allow. As long as the true solution represents the global maximum in the correlation coefficient between Fo and Fc, even if by the slimmest of margins, EPMR will eventually find it.

EPMR includes the following features:

- an option to enter fixed partial structure
- a translation search mode
- an option to bypass the evolutionary search and do only local (conjugate-gradient) optimization of the model
- independent refinement of different segments of the search model in the final conjugate-gradient minimization procedure

Usage

The program requires three input files. The first is a file containing the cell constants and space group number in the order:

```
    a b c alpha beta gamma space_group_number
```

These are free-format and can be divided between any number of lines. The program currently contains information only for the 65 enantiomorphic space groups. For rhombohedral space groups, the hexagonal (obverse), not rhombohedral setting is used and your data needs to be indexed accordingly. An example of appropriate contents for this file is:

```
    40.76 18.49 22.33 90 90.61 90 4
```

The next input file is a standard PDB format file containing the search model (orthogonal Angstrom coordinates). Any lines in this input file that are not ATOM or HETATM records (e.g. REMARK lines, etc.) are ignored. Currently only scattering factors for C, N, O, S, P and FE are incorporated into the program. Atoms of any other type are treated as carbons, except for hydrogens, which are ignored.

If the input PDB file contains multiple segment identifiers (columns 73-76, as in X-PLOR), each segment will be refined separately in the final conjugate-gradient minimization procedure. This will not affect the evolutionary search itself, where the entire contents of the file are treated as a single rigid body.

The final input file contains the observed structure factors. The only requirement is that the file has H K L Fo as the first four items on each line, separated by spaces. "fin" format files work.

These three files are all that are necessary to run the program. The command line:

```
epmr example.cel example.pdb example.hkl
```

will run the program in its default mode. In this mode, the program will search for a single molecule in the asymmetric unit. It will run the evolutionary search procedure up to ten times, or until a solution with a correlation coefficient of 0.45 is obtained. Data in the resolution range 15 to 4 Angstroms will be used in the search. The top solution found will be written to a file called 'epmr.1.best.pdb'.

The operation of the program can be controlled by a set of command line options. (If the program is run without any command line arguments, the available options will be listed.) The possible options are:

-m integer	The number of identical molecules in the asymmetric unit for which to search. The default value is 1. The flag -m2 on the command line would cause the program to search for one solution, save it as partial structure and continue searching for a second solution.
-h real_number	High-resolution limit for diffraction data used in the search (in Angstroms). The default value is 4.0 Angstroms. We do not generally recommend that this value be set to less than 5.0.
-l real_number	Low resolution limit for diffraction data (Angstroms). The default value is 15.0 Angstroms. The efficiency of the search appears to be aided slightly by the inclusion of low-resolution data. If you have accurately measured low-resolution data, you might even try a value of 25 or 30.
-p integer	The population size (number of trial solutions generated in each cycle). The default value is 300. Increasing this value beyond 300 seldom causes a dramatic improvement in search efficiency, but it never has a negative effect. If you have a fast machine you might want to try -p600, but you are likely to get more benefit from increasing the number of runs instead.
-g integer	The number of 'generations' (cycles of optimization). The default value is 50. Increasing this value has much the same effect as increasing the population size, but the degree of improvement is even less predictable. We do not generally recommend that you change this value. Setting this value to zero (-g0) will bypass the evolutionary search and feed your input model directly to the conjugate gradient optimizer. This allows you to use EPMR as a convenient rigid-body refinement program.
	The number of runs.

-n integer	The default value is 10. The program will stop before the completion of the number of runs specified here if a solution is obtained that has a correlation coefficient that exceeds a specified threshold (flag -t, below).
-t real_number	The threshold value of the correlation coefficient that indicates an acceptable solution (which will stop the run). The default value is 0.45. If the -m flag (discussed above) has a value greater than 1, the default threshold is changed to 0.3. These values generally work well for a good search model. The program will stop when a solution correlation coefficient exceeds this threshold, so if you want the program to continue for a specified number of runs no matter what, set this value to 1.0. It is best to set this value relatively high to avoid having the program stop on an incompletely converged solution. In searches for multiple molecules in the asymmetric unit, the correlation coefficient for the top solution for the first molecule is scaled by 1.5 to obtain the threshold for the next molecule. (If the new threshold is lower than the originally specified threshold, the original one is kept).
-T	Translation only mode. This will cause the program to search only translation space, keeping the orientation of the search model unchanged. This could be useful, for instance, when you have a search model that has been pre-oriented by another program or through knowledge of non-crystallographic symmetry. Note that the orientation WILL be optimized during the final rigid-body optimization after the evolutionary search, so the orientation is likely to change slightly and could change significantly during this step. This is a feature and not a bug. If you think otherwise, let us know.
-b real_number	The minimum 'bump' distance - the smallest unpenalized distance between the center of mass of a solution and that of any symmetry mates. The default value is 0.0 (no packing restrictions). This is applied to all trial solutions that are generated during the course of the search. When a solution violates this minimum distance, the correlation coefficient calculated for that solution is scaled down by the ratio of the shortest observed distance over the minimum allowed distance. In the case of searches for multiple molecules in the asymmetric unit, this also sets the minimum distance between a solution and any previously found solutions. (This applies both to previous solutions found during the run and to partial structure entered with the -s option. See the description of the -s option below for instructions on entering multiple fragments of partial structure for use with this option.) This option imposes a simple penalty on solutions that pack poorly without slowing the program significantly. This option can be helpful in some searches, but decreases the efficiency of others, particularly if a large value is used. This option appears to be less useful in single molecule searches than in searches for multiple molecules in the asymmetric unit, but definitely should be tried if you are having trouble getting a solution that packs well. It is up to you to decide what an appropriate minimum intermolecular distance should be. (Remember that it is the distance between centers of mass.) It is best to be conservative -

	the program will not search efficiently without some room to move solutions around through positions that pack poorly. In contrast to the packing restriction used in previous versions of EPMR, the packing penalty is in effect during both the evolutionary search and the final conjugate gradient minimization.
-w integer	The quantity of solutions you want written out to PDB files. The default value is 1. A '0' (zero) here means no coordinates will be written out. A '1' means only the top solution from all of the runs will be written out. A '2' means the solution obtained from each run will be written out. (The -o flag below controls the name of the output PDB files.)
-o name	The file name prefix for the output coordinate files. The default is 'epmr'. If you specify -w1 or -w2 above, the program will name the output files for each run as 'prefix'.'molecule_number'.'run_number'.pdb (e.g., epmr.1.1.pdb). The molecule number depends on how many molecules you are looking for in the asymmetric unit (option -m). The top solution from all of the runs will be 'prefix'.'molecule_number'.best.pdb. If you run multiple jobs in the same directory, you will have to use this flag to avoid writing over other solutions.
-s name	The static partial structure flag. If you have partial structure to input, include this flag and follow it with the name of the PDB file containing the correctly positioned partial structure. You can separate the partial structure into as many files as you wish and use this flag multiple times on the command line. It is only necessary to divide the partial structure up this way, however, if you are using the -b flag (see above) and are inputing multiple "pieces" of partial structure (e.g., multiple monomers). The minimum packing distance calculation will treat the partial structure within each separate file as a separate fragment.
-S	Do the initial structure factor calculation by direct summation rather than FFT. This will be MUCH slower, but if you have a very large search molecule and a workstation with little available memory, the FFT calculation may cause excessive swapping. Summation uses little memory. This only affects the single, initial FFT calculation and has no effect on the speed of the subsequent evolutionary search.
-e integer	Set the seed value for the random number generator to a specific value. This option is not necessary for normal operation of the program, but can be useful for testing purposes. If the seed is not set here, or is set to a value of zero, the seed is obtained from the system clock at run time.

The program will print some information about the settings, do the initial FFT structure factor calculation, and then start the "evolution". It will report the best score for each generation. At the end of the evolutionary search, a conjugate-gradient minimization is performed on the best scoring solution from the final generation. If there are multiple segment names in the input PDB file, the search model will first be optimized as a single rigid body, and then the separate segments will be optimized.

The final orientation for the run will be reported on a line that begins with 'Soln', followed by the run number, theta1, theta2, theta3 (Eulerian angles, defined as in X-PLOR/CNS), x translation, y translation, z translation in orthogonal Angstroms, and then the correlation coefficient and R factor. (If you intend to make use of the rotation and translation values outside of the program, they must be applied to the search model after it has been centered at the origin). You can use the UNIX command 'grep Soln log_file_name' to print the solution generated by each run. The command 'grep Soln log_file_name | sort +11 -rnb +13 -nb' will list the solutions sorted by correlation coefficient, then R-factor.

For a single molecule in the asymmetric unit, expect a correlation coefficient of 0.5 or more for a correct solution with a very good search model. For the first of two in the asymmetric unit, expect a cc near 0.30. Even poly-Ala or C-alpha models will have quite high correlation coefficients if they are accurate models. Depending on the quality of your model, however, correct solutions can have correlations well below 0.3. The best indicator of a correct solution is that it (or a symmetry-related solution) comes up repeatedly in multiple runs.

If you publish results obtained using EPMR, please cite Charles R. Kissinger, Daniel K. Gehlhaar & David B. Fogel, "Rapid automated molecular replacement by evolutionary search", Acta Crystallographica, D55, 484-491 (1999).

Note: We frequently release new versions of the program. The newest version of EPMR can be obtained from ftp://ftp.agouron.com/pub/epmr/.

We are interested in any and all questions, comments, criticisms, suggestions, and complaints about the program. Please direct these by e-mail to Chuck Kissinger at crk@agouron.com.

Acknowledgements

David Fogel (Natural Selection, Inc.) contributed numerous ideas that were essential to the successful development of this program. The structure factor calculation routines in the program were adapted from those in the XtalView software, which make use of Lynn Ten Eyck's FFT routines and were kindly made available by Duncan McRee (Syrrix Inc.).

Examples

Example 1

Search for one molecule in the asymmetric unit, perform up to 10 runs of the evolutionary search procedure (or until a cc above 0.45) is obtained, write out the best solution obtained.

```
epmr example1.cell example1.pdb example1.hkl > example1.log
```

Example 2

Search for two identical molecules in the asymmetric unit, write out all solutions, do up to 10 runs (default) for each molecule.

```
epmr -m2 -w2 example2.cel example2.pdb example2.hkl > example2.log
```

Example 3

Search for one molecule in the asymmetric unit (default), use data from 8 to 3.5 Angstroms, do up to twenty runs, write out only the top solution (default) to a file with the prefix 'example3_solution', and use static partial structure.

```
epmr -l 8.0 -h 3.5 -s example3_partial.pdb -o example3_solution -n 20
example3.cel example3.pdb example3.hkl > example3.log
```

Example 4

Search for two identical molecules in the asymmetric unit, write out all solutions, up to 10 runs (default) for each molecule. Input two fragments of partial structure. Penalize solutions that pack within 15.0 angstroms (center-to-center distance) of any symmetry mates or any partial structure that was input or generated during the run.

```
epmr -m2 -w2 -s example4_partial1.pdb -s example4_partial2.pdb -b15.0
example4.cel example4.pdb example4.hkl > example4.log
```

Example 5

Do a translational search for one molecule, use partial structure, write out all solutions, and do up to 10 runs (default) for each molecule.

```
epmr -T -w2 -s example5_a.pdb example5.cel example5_b.pdb example5.hkl >
example5.log
```